Solving the Lindblad equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = -i[H,\rho_t] + \sum_k L_k\rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}$$

Strategies and open problems

Antoine Tilloy

May 23, 2024

Quantic days, Annecy

What?

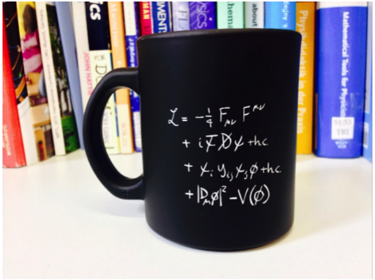
The most general **continuous Markovian** (time-local) equation for density matrices ρ:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\rho_t &= \mathcal{L}(\rho_t) \\ &= -i[H, \rho_t] + \sum_k L_k \rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}\end{aligned}$$

- $H = H^\dagger$
- $\{L_k\}$ are arbitrary

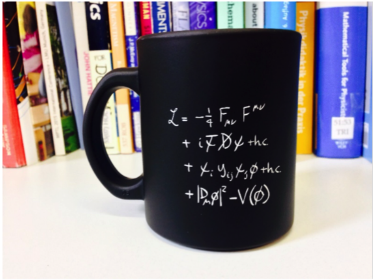
Lindblad equation as fundamental?

(So far) not a *fundamental* equation of Nature



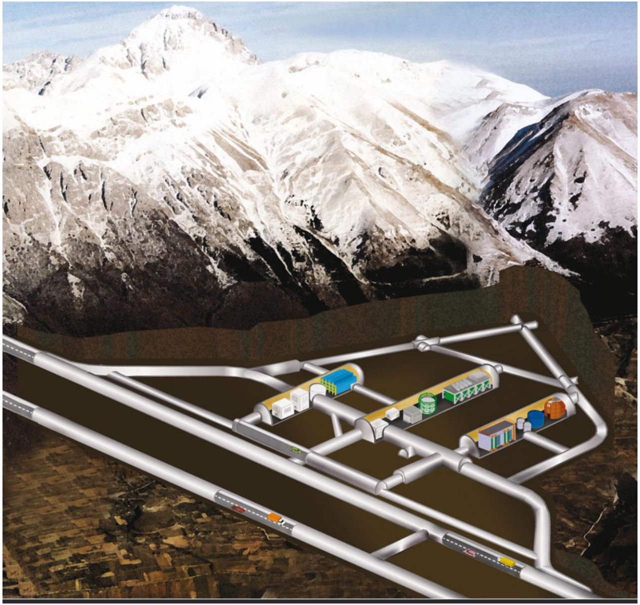
Lindblad equation as fundamental?

(So far) not a *fundamental* equation of Nature



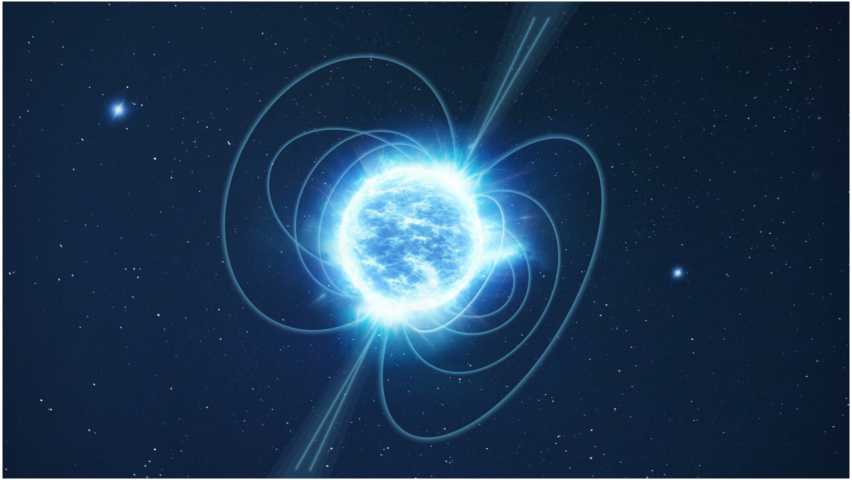
Locality + $L_k \neq 0 \quad \Longrightarrow \quad \mathrm{tr}[H\rho_t]$ increasing

Germanium does not emit X-rays spontaneously



credit Catalina Curceanu

Neutron stars do not stay hot

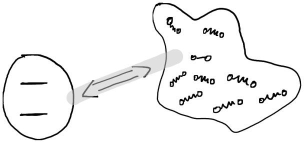


credit ESA

Derivations of the Lindblad equation

2 options

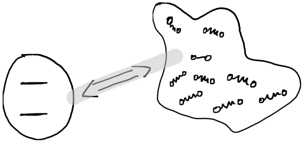
1. Coupling with a bosonic bath in some **Markov limit**



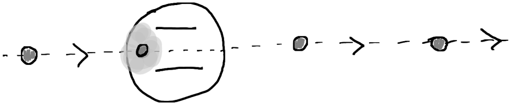
Derivations of the Lindblad equation

2 options

1. Coupling with a bosonic bath in some **Markov limit**



2. Repeated interactions in the **continuum limit** ← `easier`



Why?

- Simulate chips (or part of chips)

Why?

- Simulate chips (or part of chips)
- Solving field theories with continuous matrix product states

3 main questions

1. Real-time evolution ρ_t

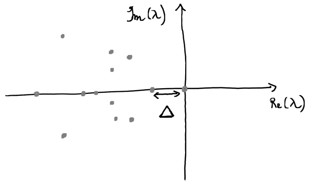
$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = \mathcal{L}(\rho_t)$$

2. Stationary state(s) ρ_0

$$\mathcal{L}(\rho_0) = 0$$

3. Spectral gap $\Delta := -\mathrm{Re}(\lambda_1)$

$$\mathcal{L}(\rho_1) = \lambda_1 \rho_1$$



Problem for Quantic

Large local Hilbert space

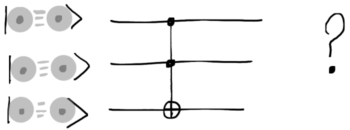
$$|\mathscr{H}| = N_c^n \text{ instead of } 2^n$$

Problem for Quantic

Large local Hilbert space

$$|\mathscr{H}| = N_c^n \text{ instead of } 2^n$$

Simulating the Toffoli gate is already a hard problem



Goal

In the “good qubit” regime:

$$N_c^n \longrightarrow C \times (2 + \varepsilon)^n$$

with arbitrarily small error

Goal

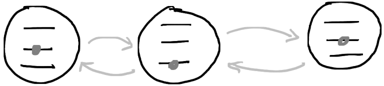
In the "good qubit" regime:

$$N_c^n \longrightarrow C \times (2+\varepsilon)^n$$

with arbitrarily small error

- Reasonable from complexity theory $\mathsf{BQP} \neq \mathsf{BPP}$
- With enough noise $N_c^n \longrightarrow \mathrm{poly}(n)$ `[for supremacy]`

The basics: “small” Hilbert space case



The one and only trick: $\mathcal{L}$ sparsity

Typical Lindbladian:

$$\mathcal{L}(\rho_t) := -i[H, \rho_t] + \sum_{k=1}^{r} L_k \rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}$$

2 sources of sparsity:

The one and only trick: $\mathcal{L}$ sparsity

Typical Lindbladian:

$$\mathcal{L}(\rho_t) := -i[H, \rho_t] + \sum_{k=1}^{r} L_k \rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}$$

2 sources of sparsity:

- Few generators $r \ll N$ – $\mathcal{L}(\rho)$ cost $N^4 \to rN^3$

The one and only trick: $\mathcal{L}$ sparsity

Typical Lindbladian:

$$\mathcal{L}(\rho_t) := -i[H, \rho_t] + \sum_{k=1}^{r} L_k \rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}$$

2 sources of sparsity:

- Few generators $r \ll N$ – $\mathcal{L}(\rho)$ cost $N^4 \to rN^3$
- Almost diagonal L_k – $\mathcal{L}(\rho)$ cost $\to rN^2$ (optimal)

The one and only trick: $\mathcal{L}$ sparsity

Typical Lindbladian:

$$\mathcal{L}(\rho_t) := -i[H, \rho_t] + \sum_{k=1}^{r} L_k \rho_t L_k^\dagger - \frac{1}{2}\left\{L_k^\dagger L_k, \rho_t\right\}$$

2 sources of sparsity:

- Few generators $r \ll N$ – $\mathcal{L}(\rho)$ cost $N^4 \to rN^3$
- Almost diagonal L_k – $\mathcal{L}(\rho)$ cost $\to rN^2$ (optimal)

$\implies$ See $\mathcal{L}$ as a *function* not a matrix

Real-time evolution – small $\mathscr{H}$

Find $t \mapsto \rho_t$ for

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = \mathcal{L}(\rho_t)$$

with $\mathcal{L}$ time independent

Real-time evolution – small $\mathscr{H}$

Find $t \mapsto \rho_t$ for

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = \mathcal{L}(\rho_t)$$

with $\mathcal{L}$ time independent

Solutions

1. Exponentiate $\rho_t = \mathrm{e}^{t\mathcal{L}} \cdot \rho_{t=0}$ (with Krylov)
2. Solve the linear ordinary differential equation (ODE)

Real-time evolution – small $\mathscr{H}$

Find $t \mapsto \rho_t$ for

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = \mathcal{L}(\rho_t)$$

with $\mathcal{L}$ time independent

Solutions

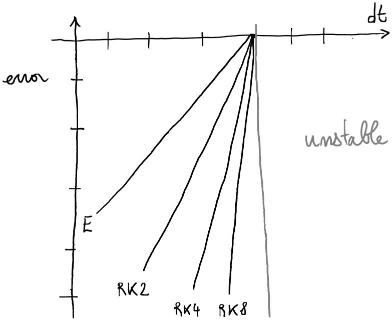
1. Exponentiate $\rho_t = \mathrm{e}^{t\mathcal{L}} \cdot \rho_{t=0}$ (with Krylov)
2. Solve the linear ordinary differential equation (ODE)

Exponentiation better for few points, ODE for full trajectory

Solving a (linear) ordinary differential equation

From Rémi's talk:

- Higher-order Runge-Kutta: numerically exact or blows up
- Low-order Rouchon: doesn't blow up



Stationary state

Finding ρ_0 such that $\mathcal{L}(\rho_0) = \rho_0$

Strategies:

1. Do real time evolution for a long time
2. Solve the linear problem $\mathcal{L}(\rho_0) = 0$
3. Find the lowest eigenvector of $\mathcal{L}$

Stationary state

Finding ρ_0 such that $\mathcal{L}(\rho_0) = \rho_0$

Strategies:

1. Do real time evolution for a long time
2. Solve the linear problem $\mathcal{L}(\rho_0) = 0$
3. Find the lowest eigenvector of $\mathcal{L}$

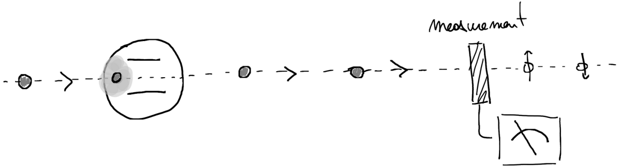
Cost: always rN^3 for iterative methods (N^6 for strictly exact)

Remark: Stochastic trajectory method

Quantum trajectory method

For ρ_t solution of Lindblad, $\exists$ $|\psi_t\rangle$ obeying **stochastic non-linear** equation such that:

$$\rho_t = \mathbb{E}\Big[|\psi_t\rangle\langle\psi_t|\Big]$$

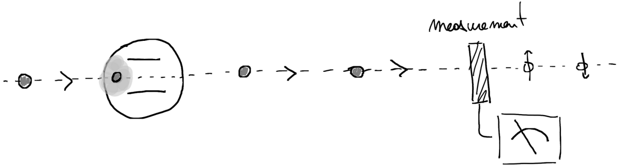


Remark: Stochastic trajectory method

Quantum trajectory method

For ρ_t solution of Lindblad, $\exists$ $|\psi_t\rangle$ obeying **stochastic non-linear** equation such that:

$$\rho_t = \mathbb{E}\Big[|\psi_t\rangle\langle\psi_t|\Big]$$



Example: photo-detection / homodyne-detection with $\eta = 1$

- Requires less memory
- Obliterates precision

Main interest: leveraging approximate algorithms made for pure states

With bosons

Bosonic Hilbert space:

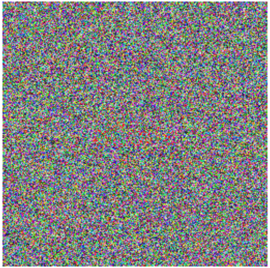
$$\mathscr{H} = L^2(\mathbb{R}) = \mathrm{span}\{a^{\dagger k}|0\rangle\}$$

Need some truncation / compression

Compression

No free lunch

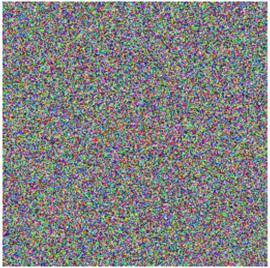
There will always be situations in which $L^2(\mathbb{R})$ cannot efficiently be compressed



Compression

No free lunch

There will always be situations in which $L^2(\mathbb{R})$ cannot efficiently be compressed



A compression method that is not bad at something, cannot be good at anything

Fock space truncation

Choose $a^{\dagger k}|0\rangle$ for $0 \leqslant k \leqslant N_c$
In real-space: Hermite functions

$$\psi_n(x) = H_n(x) \exp(-x^2/2)$$

Fock space truncation

Choose $a^{\dagger k}|0\rangle$ for $0 \leqslant k \leqslant N_c$
In real-space: Hermite functions

$$\psi_n(x) = H_n(x)\exp(-x^2/2)$$

Advantages:

- Error $\propto \mathrm{e}^{-CN_c}$, $\rightarrow$ **numerically exact**
- L_k often sparse in this basis

Fock space truncation

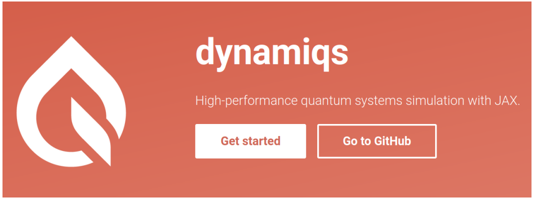
No free lunch:

$$N_c \propto |\alpha|^2$$

Good cat qubits require **larger** Hilbert space

State of the art: Dynamiqs

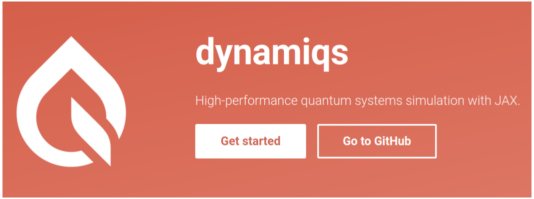
The previous tricks $+$ smart use of hardware $+$ autodiff $+$ batching $\rightarrow$ `dynamiqs`



Close to the numerical optimum

State of the art: Dynamiqs

The previous tricks + smart use of hardware + autodiff + batching → `dynamiqs`



Close to the numerical optimum

`QuantumOptics.jl` also good (without seemless autodiff, batching, GPU)

Technical open problems

Exact stationary state

Find ρ_0 *exactly* at cost rN^3.
Currently:

- rN^3 approximate fixed-point algorithms (e.g. Krylov)
- N^6 for linear solve

Technical open problems

Exact stationary state

Find ρ_0 *exactly* at cost rN^3.
Currently:

- rN^3 approximate fixed-point algorithms (e.g. Krylov)
- N^6 for linear solve

Precise time-dependent real-time evolution

Find a method more precise than Runge-Kutta for time-dependent real-time dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_t = \mathcal{L}_t(\rho_t)$$

using the fact that $\mathcal{L}_t$ is *linear*

Summary for small Hilbert space

Lindblad for bosons = partial differential equations in $2n_{\text{bosons}} + 1$ dimensions

Good

- Solved **numerically exactly** at cost $rN^2 \propto N_c^{2n_{\text{bosons}}}$!
- It is implemented in user friendly libraries
- It is easy to implement manually to test

Summary for small Hilbert space

Lindblad for bosons = partial differential equations in $2n_{\text{bosons}} + 1$ dimensions

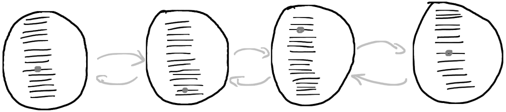
Good

- Solved **numerically exactly** at cost $rN^2 \propto N_c^{2n_{\text{bosons}}}$!
- It is implemented in user friendly libraries
- It is easy to implement manually to test

Bad

- For cats $N \sim (4|\alpha|^2)^{n_{\text{cats}}}$
- Compute and memory cost $\propto |2\alpha|^{4n_{\text{cats}}}$

Advanced: Large Hilbert space



Analytic solutions: lucky cases

Gaussian states

Equivalently:

- ρ is a Gaussian in configuration space
- ρ has a Gaussian Wigner function
- $\rho = \exp(\text{quadratic in } a, a^\dagger)$

$\implies$ ρ characterized by $\mathrm{tr}(a\rho)$, $\mathrm{tr}(aa\rho)$ and $\mathrm{tr}(a^\dagger a\rho)$

Analytic solutions: lucky cases

Gaussian states

Equivalently:

- ρ is a Gaussian in configuration space
- ρ has a Gaussian Wigner function
- $\rho = \exp(\text{quadratic in } a, a^\dagger)$

$\implies$ ρ characterized by $\mathrm{tr}(a\rho)$, $\mathrm{tr}(aa\rho)$ and $\mathrm{tr}(a^\dagger a\rho)$

Quadratic Lindbladian

1. H at most quadratic in $a, a^\dagger$
2. A_k at most linear in $a, a^\dagger$

$\mathcal{L}$ quadratic $\implies$ exactly solvable with Gaussian states

Analytic solutions: lucky cases

Gaussian states

Equivalently:

- ρ is a Gaussian in configuration space
- ρ has a Gaussian Wigner function
- $\rho = \exp(\text{quadratic in } a, a^\dagger)$

$\Longrightarrow$ ρ characterized by $\mathrm{tr}(a\rho)$, $\mathrm{tr}(aa\rho)$ and $\mathrm{tr}(a^\dagger a\rho)$

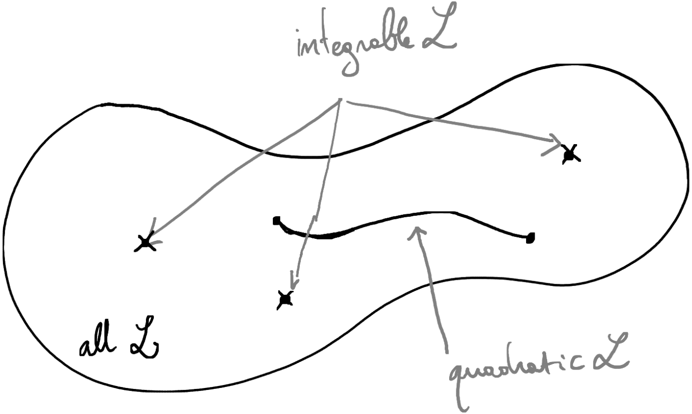
Quadratic Lindbladian

1. H at most quadratic in $a, a^\dagger$
2. A_k at most linear in $a, a^\dagger$

$\mathcal{L}$ quadratic $\Longrightarrow$ exactly solvable with Gaussian states

Small number of other exactly solvable (integrable) cases found by Tomaz Prosen

Exact solutions are too rare to be really useful



An idea?

Can't we find a better basis??

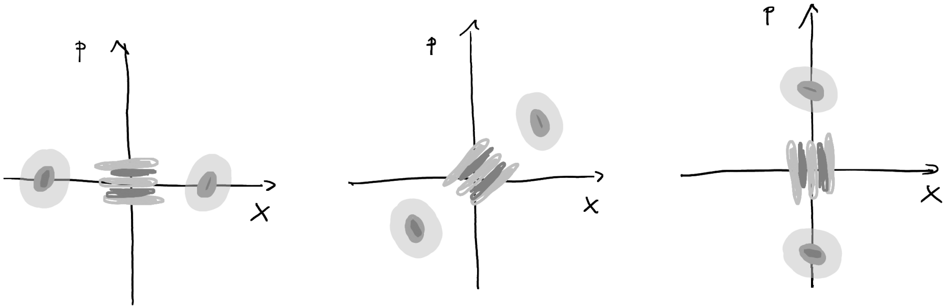
An idea?

Can't we find a better basis??

NO!

Linear stability should be abandonned

No "best" plane to approximate a sphere



Submanifold for local qubits

Many interesting subspaces $\mathcal{M} \subset \mathscr{H}$ are not vector spaces:

- coherent states
- Gaussian states
- ρ with fixed rank
- "Dynamical shifted Fock"

Submanifold for local qubits

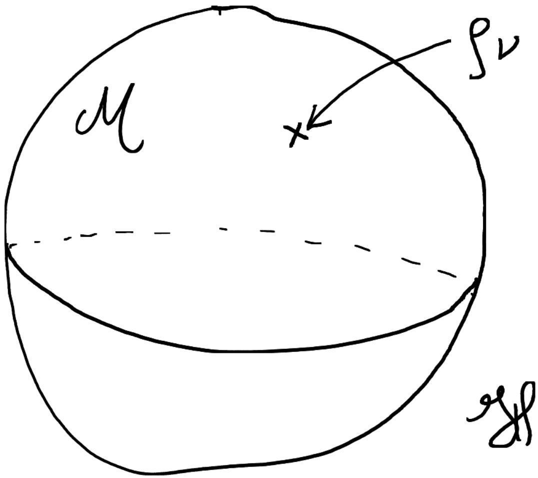
Many interesting subspaces $\mathcal{M} \subset \mathscr{H}$ are not vector spaces:

- coherent states
- Gaussian states
- ρ with fixed rank
- "Dynamical shifted Fock"

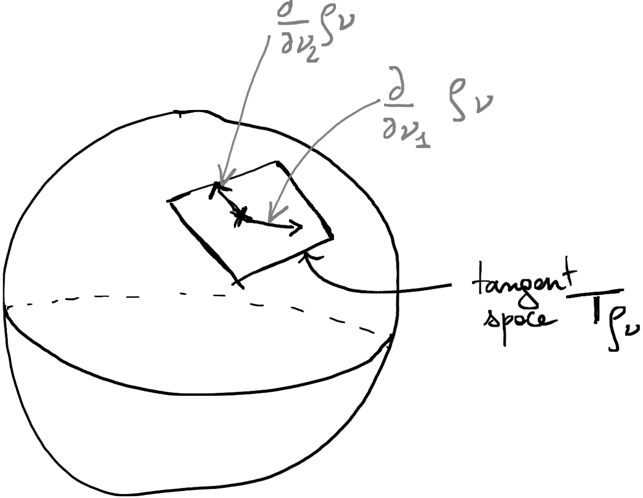
Example

$$\rho_\nu = |\psi_\nu\rangle\langle\psi_\nu| \quad \text{with} \quad \psi_\nu(x) = \exp(-\nu_1 x^2 - \nu_2 x^4)$$

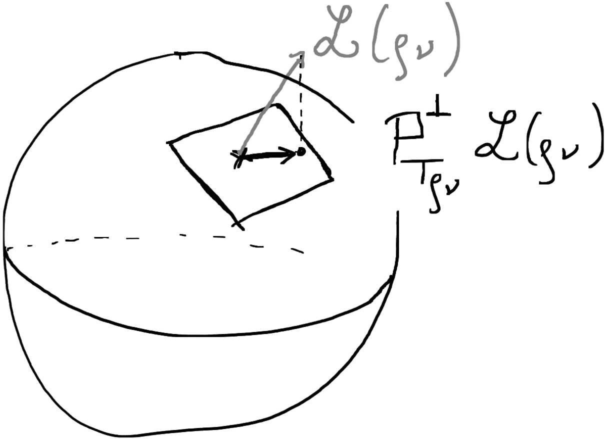
Time-dependent variational principle (TDVP)



Time-dependent variational principle (TDVP)



Time-dependent variational principle (TDVP)



Time-dependent variational principle (TDVP)

McLachlan TDVP = minimize the error

$$\text{error} = \left\| \frac{\mathrm{d}}{\mathrm{d}t} \rho(\nu_t) - \mathcal{L}[\rho(\nu_t)] \right\|^2 \tag{1}$$

for convenience, take the Frobenius norm

$$\|A\| = \sqrt{\mathrm{tr}(A^\dagger A)} \tag{2}$$

TDVP

$$\frac{\partial}{\partial \frac{\mathrm{d}}{\mathrm{d}t}\nu_j} \left\| \frac{\mathrm{d}}{\mathrm{d}t}\rho(\nu_t) - \mathcal{L}[\rho(\nu_t)] \right\|^2 = 0$$

TDVP

$$\frac{\partial}{\partial \frac{\mathrm{d}}{\mathrm{d}t}\nu_j} \left\| \frac{\mathrm{d}}{\mathrm{d}t}\rho(\nu_t) - \mathcal{L}[\rho(\nu_t)] \right\|^2 = 0$$

The time derivative of ρ is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(\nu_t) = \frac{\partial \rho}{\partial \nu_j}\frac{\mathrm{d}\nu_j}{\mathrm{d}t} = \frac{\partial \rho}{\partial \nu_j}\frac{\mathrm{d}}{\mathrm{d}t}\nu_j$$

thus

TDVP

$$\frac{\partial}{\partial \frac{d}{dt}\nu_j} \left\| \frac{d}{dt}\rho(\nu_t) - \mathcal{L}[\rho(\nu_t)] \right\|^2 = 0$$

The time derivative of ρ is:

$$\frac{d}{dt}\rho(\nu_t) = \frac{\partial \rho}{\partial \nu_j}\frac{d\nu_j}{dt} = \frac{\partial \rho}{\partial \nu_j}\frac{d}{dt}\nu_j$$

thus

$$\frac{d}{dt}\nu_j \operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j}\right)^\dagger \frac{\partial \rho}{\partial \nu_k} + \text{h.c.}\right] - \operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j}\right)^\dagger \mathcal{L}(\rho) + \text{h.c.}\right] = 0$$

TDVP

$$\frac{\partial}{\partial \frac{d}{dt}\nu_j} \left\| \frac{d}{dt}\rho(\nu_t) - \mathcal{L}[\rho(\nu_t)] \right\|^2 = 0$$

The time derivative of ρ is:

$$\frac{d}{dt}\rho(\nu_t) = \frac{\partial \rho}{\partial \nu_j}\frac{d\nu_j}{dt} = \frac{\partial \rho}{\partial \nu_j}\frac{d}{dt}\nu_j$$

thus

$$\frac{d}{dt}\nu_j \operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j}\right)^\dagger \frac{\partial \rho}{\partial \nu_k} + \text{h.c.}\right] - \operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j}\right)^\dagger \mathcal{L}(\rho) + \text{h.c.}\right] = 0$$

Hence the dynamics is $\frac{d}{dt}\nu_j = [g]^{-1}_{jk}\, u_k$ with

$$g_{jk} = \Re\mathrm{e}\left\{\operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j}\right)^\dagger \frac{\partial \rho}{\partial \nu_k}\right]\right\}$$

$$u_k = \Re\mathrm{e}\left\{\operatorname{tr}\left[\left(\frac{\partial \rho}{\partial \nu_k}\right)^\dagger \mathcal{L}(\rho)\right]\right\}$$

Requirements

Solve the non-linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\nu_j = [g]^{-1}_{jk}\, u_k$$

with Runger-Kutta (or other)

Requirements

Solve the non-linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\nu_j = [g]^{-1}_{jk}\, u_k$$

with Runger-Kutta (or other)

Only need:

$$g_{jk} = \Re\mathrm{e}\left\{ \mathrm{tr}\left[\left(\frac{\partial \rho}{\partial \nu_j} \right)^\dagger \frac{\partial \rho}{\partial \nu_k} \right] \right\}$$

$$u_k = \Re\mathrm{e}\left\{ \mathrm{tr}\left[\left(\frac{\partial \rho}{\partial \nu_k} \right)^\dagger \mathcal{L}(\rho) \right] \right\}$$

Requirements

Solve the non-linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\nu_j = [g]^{-1}_{jk}\, u_k$$

with Runger-Kutta (or other)

Only need:

$$g_{jk} = \Re\mathrm{e}\left\{\mathrm{tr}\left[\left(\frac{\partial\rho}{\partial\nu_j}\right)^{\dagger}\frac{\partial\rho}{\partial\nu_k}\right]\right\}$$

$$u_k = \Re\mathrm{e}\left\{\mathrm{tr}\left[\left(\frac{\partial\rho}{\partial\nu_k}\right)^{\dagger}\mathcal{L}(\rho)\right]\right\}$$

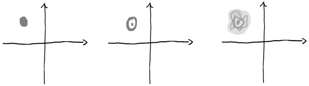
4 levels

- Exactly computable (Shifted Fock / MPS / Quantics) → `best`
- Well approximable (2d tensor networks)
- Can be directly sampled with Monte Carlo (some recurrent neural nets)
- Requires Markov Chain Monte Carlo for approximate sampling (generic neural nets)

Local strategy I: dynamical shifted Fock

The shifted Fock ladder (vector space):

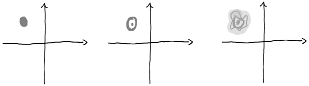
$$\mathcal{M}_\alpha^{N_c} = \text{span}\{a^{\dagger k}|\alpha\rangle,\ \ k \leqslant N_c\} = \text{span}\{D(\alpha)|k\rangle,\ \ k \leqslant N_c\}$$

$=$ 

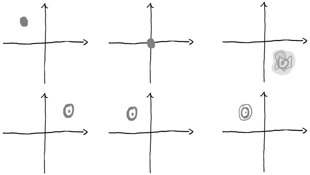
Local strategy I: dynamical shifted Fock

The shifted Fock ladder (vector space):

$$\mathcal{M}_\alpha^{N_c} = \mathrm{span}\{a^{\dagger k}|\alpha\rangle,\ \ k \leqslant N_c\} = \mathrm{span}\{D(\alpha)|k\rangle,\ \ k \leqslant N_c\}$$

$$=$$ 

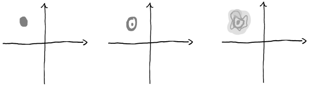
The "dynamical" shifted Fock ladder (manifold):

$$\mathcal{M}^{N_c} = \{|\psi\rangle \in \mathcal{M}_\alpha,\ \alpha \in \mathbb{C}\} =$$ 

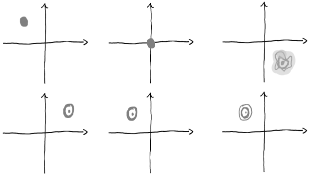
Local strategy I: dynamical shifted Fock

The shifted Fock ladder (vector space):

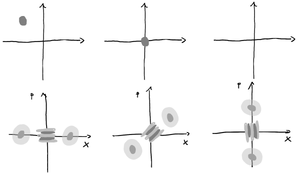
$$\mathcal{M}_{\alpha}^{N_c} = \mathrm{span}\{a^{\dagger k}|\alpha\rangle, \quad k \leqslant N_c\} = \mathrm{span}\{D(\alpha)|k\rangle, \quad k \leqslant N_c\}$$

$$=$$ 

The "dynamical" shifted Fock ladder (manifold):

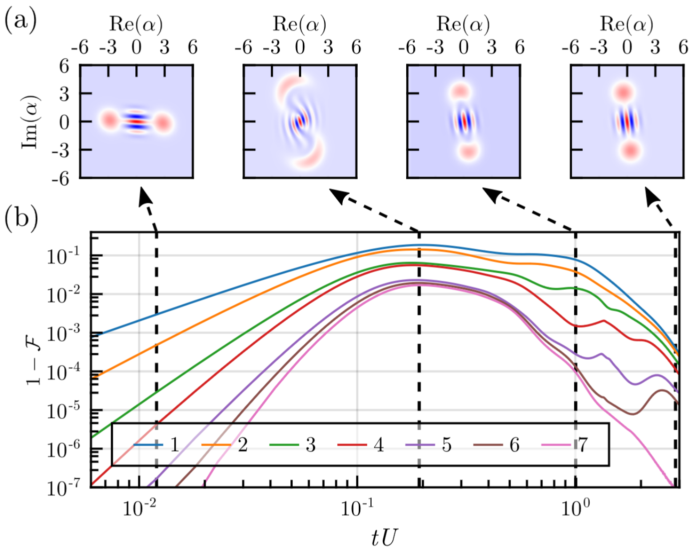
$$\mathcal{M}^{N_c} = \{|\psi\rangle \in \mathcal{M}_{\alpha}, \ \alpha \in \mathbb{C}\} =$$ 

The multi "dynamical" shifted Fock ladder (manifold):

$$\mathcal{M}_{(n)}^{N_c} = \left\{ |\psi\rangle = \sum_{i=1}^{n} |\psi_i\rangle, \ |\psi_i\rangle \in \mathcal{M} \right\} =$$ 

Dynamical shifted Fock

Proposed by Schlegel, Minganti, Savona in 2306.13708



Dynamical shifted Fock: pros and cons

The Good

- Good intuition for compression
- Works as intended $N_c \to 2 + \varepsilon$

The Bad

- Sparsity lost
- Potential ill-conditioning (fixable)
- Cannot destroy the blobs too much

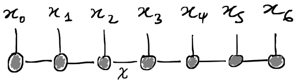
Likely not faster for 1 mode

Quantics

Idea of Adrien Moulinas and Xavier Waintal

Write ψ as an MPS in binary:

$$\psi(x_0.x_1x_2x_3x_4x_5x_6) = \langle A(0)_{x_0}|A(1)_{x_1}A(2)_{x_2}A(3)_{x_3}A(4)_{x_4}A(5)_{x_5}|A(6)_{x_6}\rangle$$

$=$ 

with

- $A(0)_0, A(0)_1$ and $A(6)_0, A(6)_1$ vectors $\in \mathbb{C}^\chi$
- $A(1)_0, A(1)_1, \cdots A(5)_0, A(5)_1$ matrices $\in \mathbb{C}^\chi \otimes \mathbb{C}^\chi$

Quantics: pros and cons

Still to be tested, but:

The Good (promising):

- Leverages powerful tensor network routines (could be fast!)
- Can resolve arbitrarily small scales

The Bad (worrying):

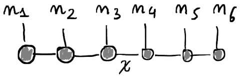
- Not super clear when it should work intuitively
- Entanglement in scale??

Matrix product state like approaches

Mainly 3 options (tried at Eviden on slightly different problems)

1. Matrix Product States for stochastic ψ

$$\psi(n_1, \cdots, n_6) =$$

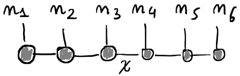


Matrix product state like approaches

Mainly 3 options (tried at Eviden on slightly different problems)

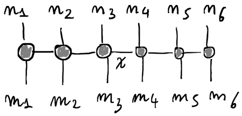
1. Matrix Product States for stochastic ψ

$$\psi(n_1, \cdots, n_6) =$$



2. Matrix Product Operators for ρ

$$\rho(n_1, \cdots, n_6, m_1, \cdots, m_6) =$$

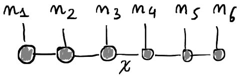


Matrix product state like approaches

Mainly 3 options (tried at Eviden on slightly different problems)

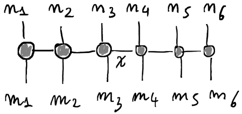
1. **1.** Matrix Product States for stochastic ψ

$$\psi(n_1, \cdots, n_6) =$$



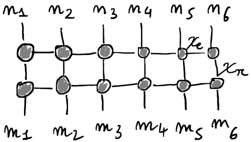
2. **2.** Matrix Product Operators for ρ

$$\rho(n_1, \cdots, n_6, m_1, \cdots, m_6) =$$



3. **3.** Matrix Product Density Operator for ρ (aka *purification*)

$$\rho(n_1, \cdots, n_6, m_1, \cdots, m_6) =$$



MPS approaches: intuition

Hope: $\chi \propto (2+\varepsilon)^n$ (not much inter-cat entanglement)

- No local physical space compression
- Asymptotic cost $N_c^4 * (2+\varepsilon)^n$

No gain on $\leqslant 2$ modes

Strategy

Small exact simulations: memory cost N_c^{2n}

- Mostly Solved
- Few open problems (very high order Rouchon, cheaper exact stationary states)
- No huge gain expected unless ultra-stiff
- Well implemented

Large simulations: target is memory cost $C \times (2 + \varepsilon)^{2n}$

- Local compression way (Dynamically shifted Fock, Quantics)
- Global compression way (MPS,MPO,MPDO)
- Nothing well implemented yet

Do not ask for performance increase at the beginning for small systems!